**CHAPTER four**

**System Implementation, Result and Discussion**

**4.1 Introduction**

The implementation details of the proposed Nigeria Blockchain Budget Management System (NBBMS) is comprehensively explained in this chapter. The choice of programming language, development process, coding section and the integrated development environment (IDE) used are also presented. Finally, the system is tested and evaluated in respect to the existing traditional Budget management system.

**4.2 Choice of programming language**

The proposed Nigeria Budget Blockchain budget Management system (NBBMS) is developed using Django (Python Web Framework). Django is a web-based framework for python developer, its enable fast development of standalone website. It’s the programming language of choice in this thesis due to it robustness, flexibility, scalability, security and easy integration with SQL relational databased such as;

1. SQLite database.
2. Oracle RDBMS.
3. Postgres and lot more.

**4.3 System Requirement**

The proposed system development process can only be achieved with two main classes of system requirement, this includes the hardware requirement and the Software requirement. The two main perspective of system requirement for this project, is further explained in the following sub section

**4.3.1. Hardware Requirement**

the minimum recommended hardware requirement for developing and running the developed system includes Minimum of;

1. 4 Gigabytes of Random Access Memory (4GB)
2. Window operating system (window 7 version)
3. 100 GB Hard Disk Drive
4. 2.0 GHz processor speed

However, the proposed system is developed using

1. 8 GB of RAM
2. Window 10 Operating System
3. 500 GB Hard Disk Drive
4. 2.60 GHz processor speed

**Table 4.1 Requirement Summary**

|  |  |  |  |
| --- | --- | --- | --- |
| **S/N** | **Hardware Specification** | **Min Specification** | **Used Specification** |
| **1.** | RAM | 4GB | 8GB |
| **2.** | Operation System (window) | Window 7 | Window 10 |
| **3.** | Hard Disk Drive (HDD) | 100GB | 500GB |
| **4.** | CPU | 2.0GHz | 2.60GHz |

**4.3.2. Software Requirement**

the software requirement includes the development environment, programming language interpreter, web development framework dependency.

1. **Python Interpreter**: the python interpreter is essential in the development process, this enables the machine to understand and translate python syntax. This research thesis considered installing version 3.9 Python interpreter.
2. **Django**: the Django Framework is a high-level web development framework in python, its essential in this development process because it offer a secure and rapid maintainable website.
3. **PostgreSQL and Admin:** this provide databased server and Admin panel to manage user data, transaction information and the likes. The PostgreSQL is adopted because it works well with the Django web framework.
4. **PSYCOPG:** the PSYCOPGis an important dependency which is required to connect the proposed system to the database management system.
5. **VS code:** The Vs Code is an integrated development environment for written source codes. Its compatible with many programming languages with python language included. However, it provides a nice graphical user interface environment for convenient source code written.

**4.3.2.1. Functional Requirement**

the Basic functional requirement of the propose system includes:

1. The System should be able to generate cryptography identity for new users
2. The System should be able to secure transactions initiated by users
3. The System should be able to perform basic validation on transaction, and authentication of user identity.
4. The system is capable of visualizing user information, historical transactions, and blocks.

**4.3.2.2. Non-Functional Requirement**

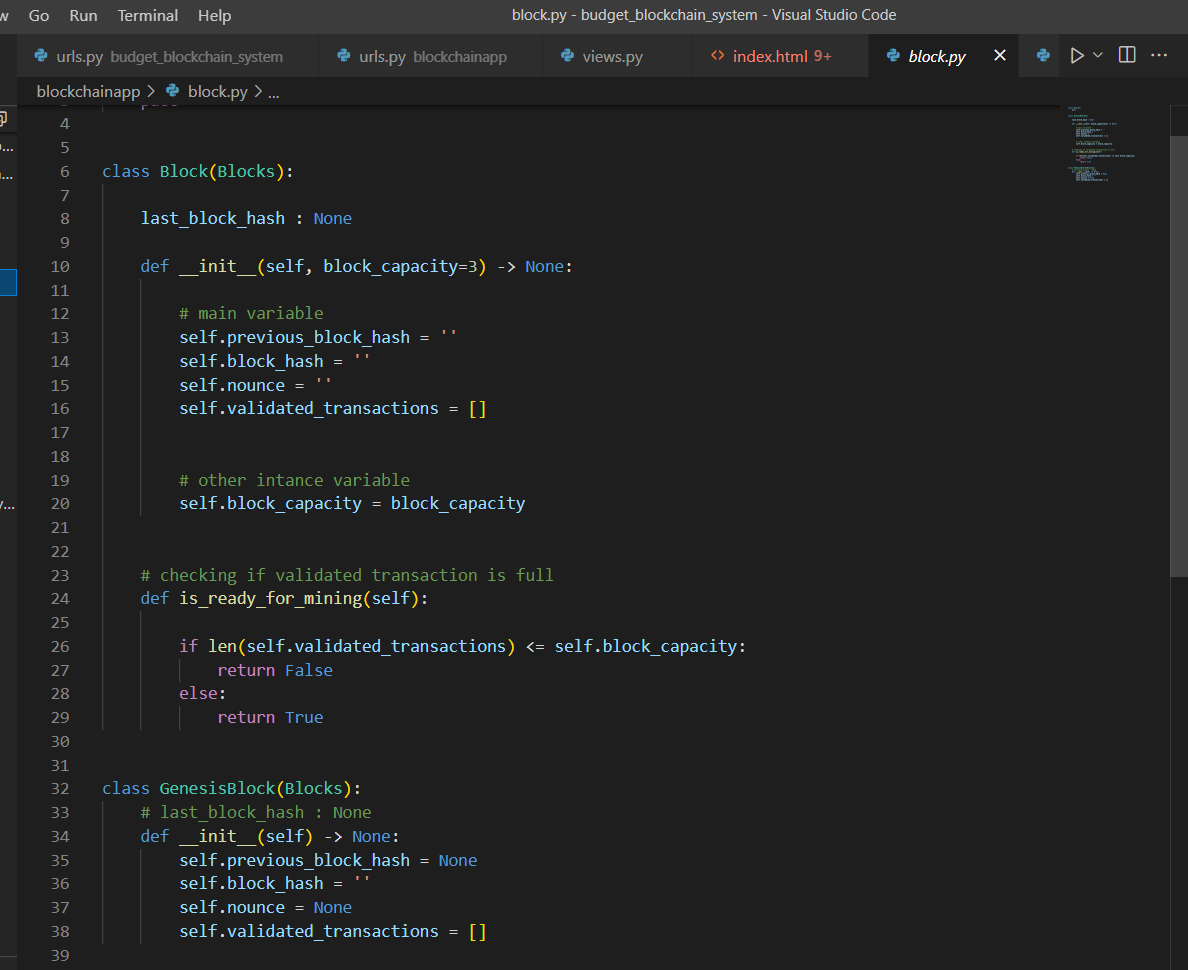
the non-functional requirement of the proposed NBBMS includes;

1. A friendly and easy user interface
2. Availability of service at all time
3. The system should be transparent
4. The system should be secure, immutable and reliable
5. Decentralize based system

**4.4. Implemented Budget Blockchain System**

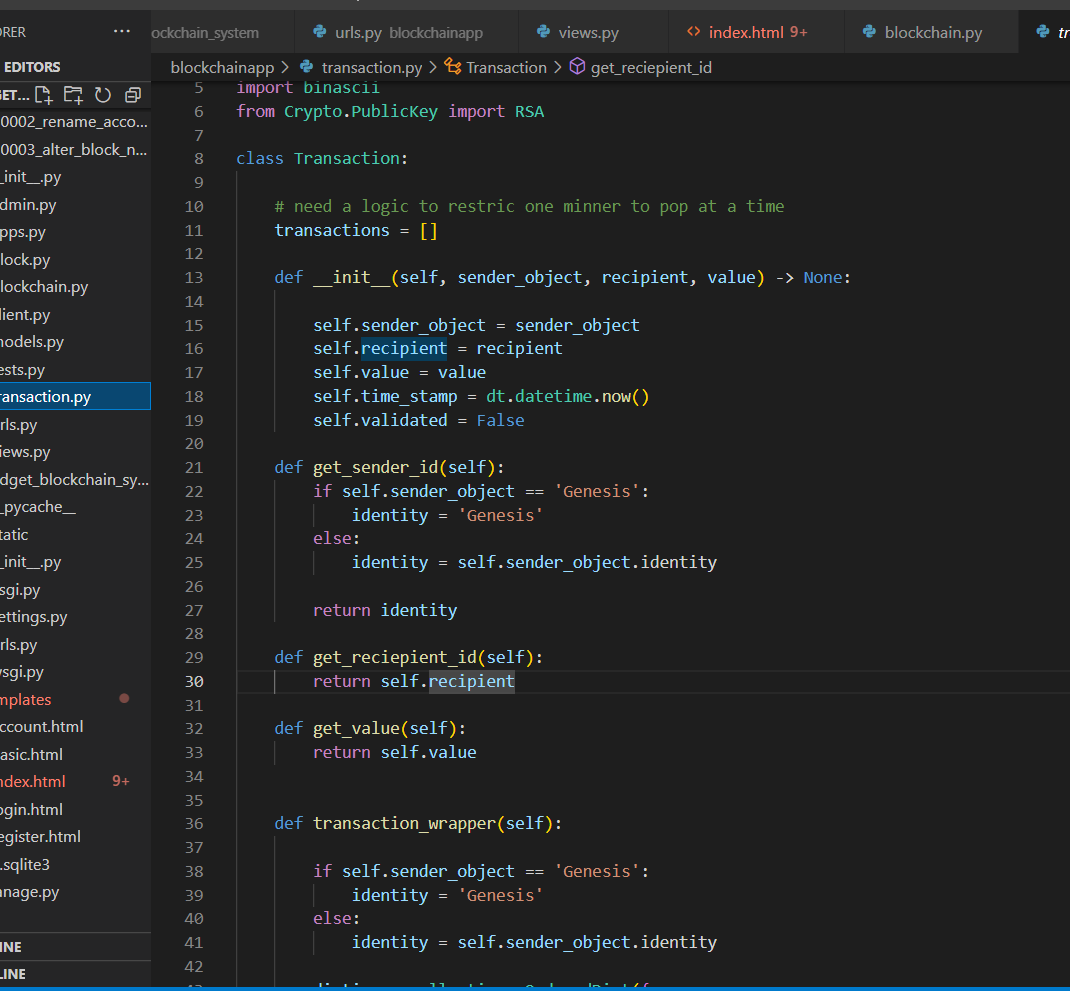
This sub section describes the various development stage of the propose Nigeria Blockchain Budget Management system. The development process comprises of Django Model Creation, Object Relational Mapping, URL Mapping, Database Management, and the User interface Designs etc.

**4.4.1 The Python Model Classes (VS code Snippets)**



**Fig 4.1 The Genesis and Subsequent Block Class Model**

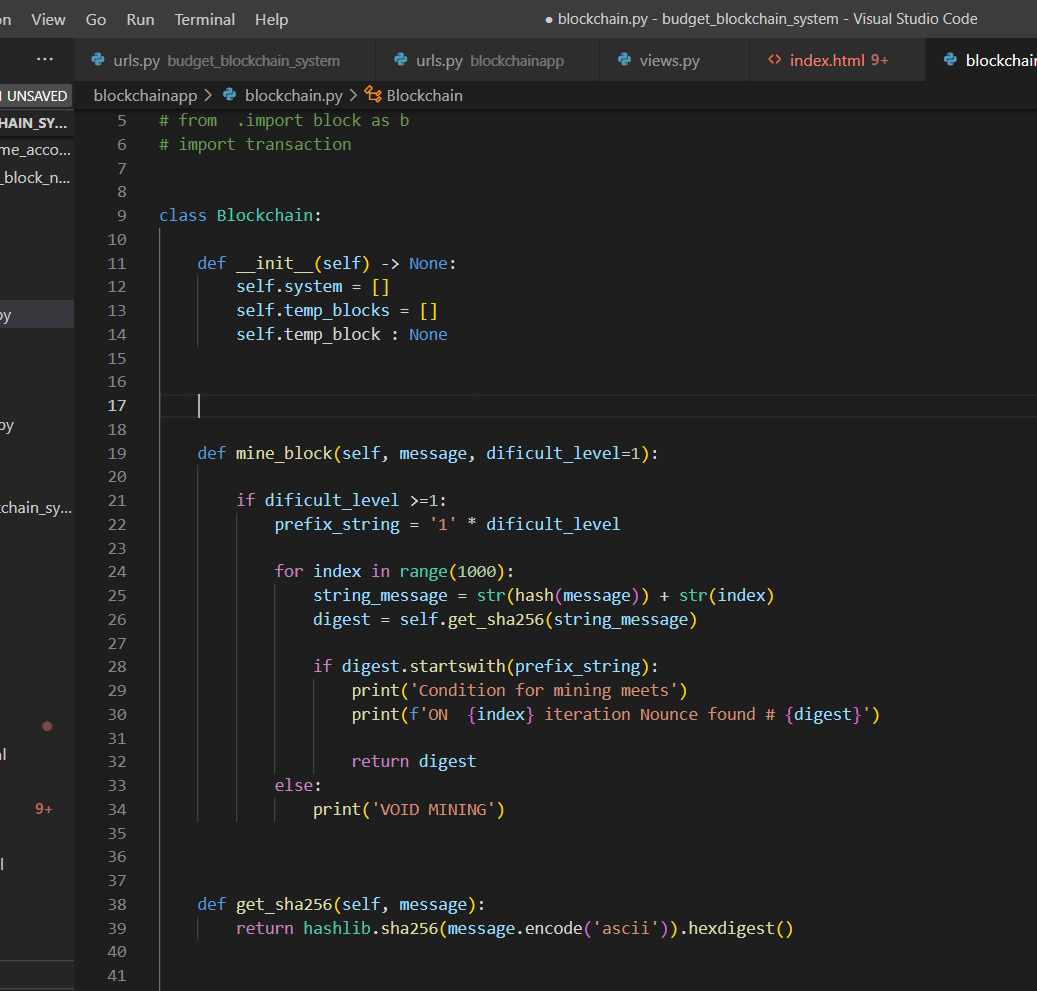
The Fig 4.1 illustrate the code section for implementing the Genesis block and the Subsequent Block in the blockchain system. The Genesis block denote the first block to be created in the blockchain system. based on the sturdy the ‘Accountant General of Nigeria Federation’ denote the Genesis block in the proposed NBBMS system. According to the diagrammatic illustration, the block holds and instance variables of previous hash block, nonce value, block hash, and list of validated transactions. However, the mining capacity of each block is implemented using the ‘*is\_ready\_for\_mining*’ helper method.



**Fig 4.2 The Transaction Class Implementation**

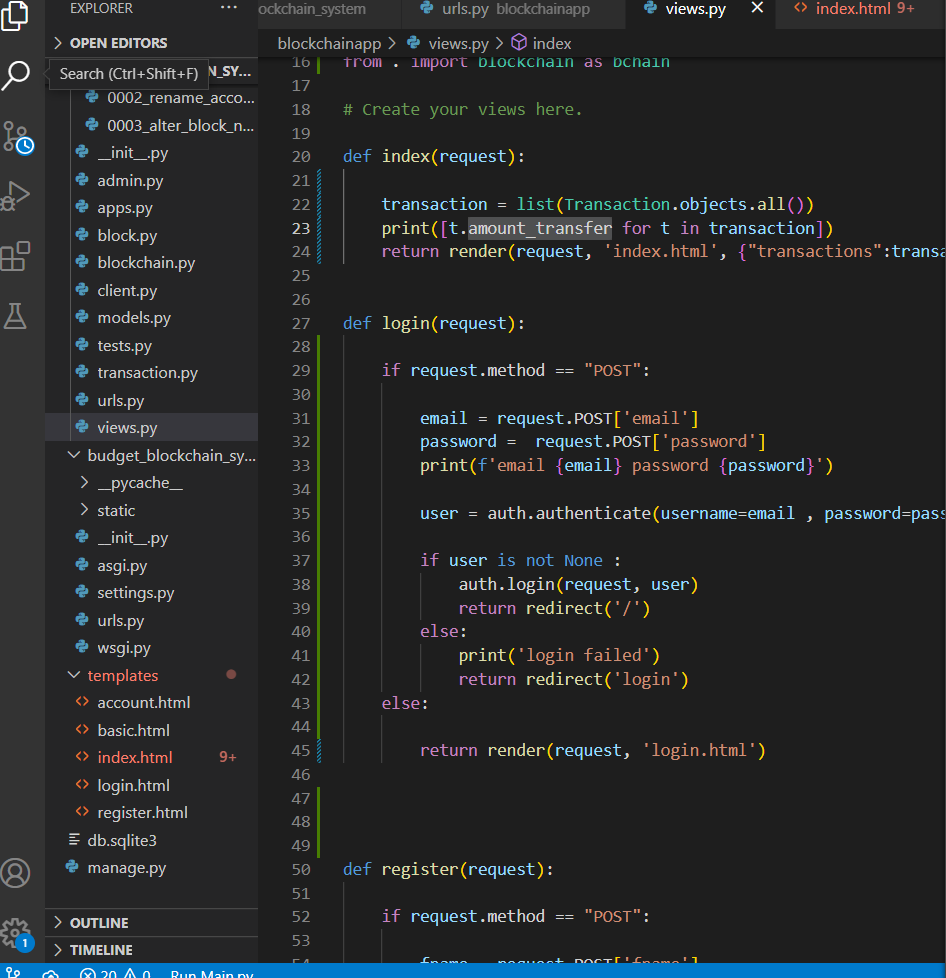
Based on the illustration on fig 4.2 show the code section for implementing the Transaction model. The Transaction model hold information such as the sender, receiver instance and the exchange value. In addition, the time stamp at which the transaction take place with some helper methods such as; get sender identity, get recipients identity, sign transaction and the likes.

Fig 4.3 describe the blockchain implement model within the VS code environment. The instance of the Blockchain class hold information of all Blocks, and the helper method for simulating the mining procedure of a blockchain. The mining processing is implemented with a difficult level of 1 by default.



**Fig 4.3 The Blockchain Class Implementation**

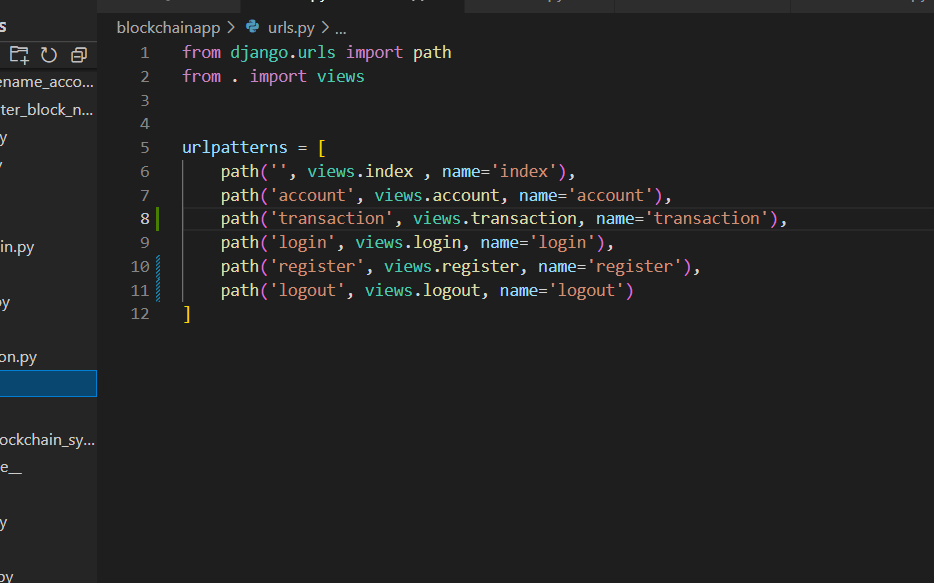
**4.4.2 The Django Framework (Code Snippet)**



**Fig 4.4 The Django Framework Request Handlers**

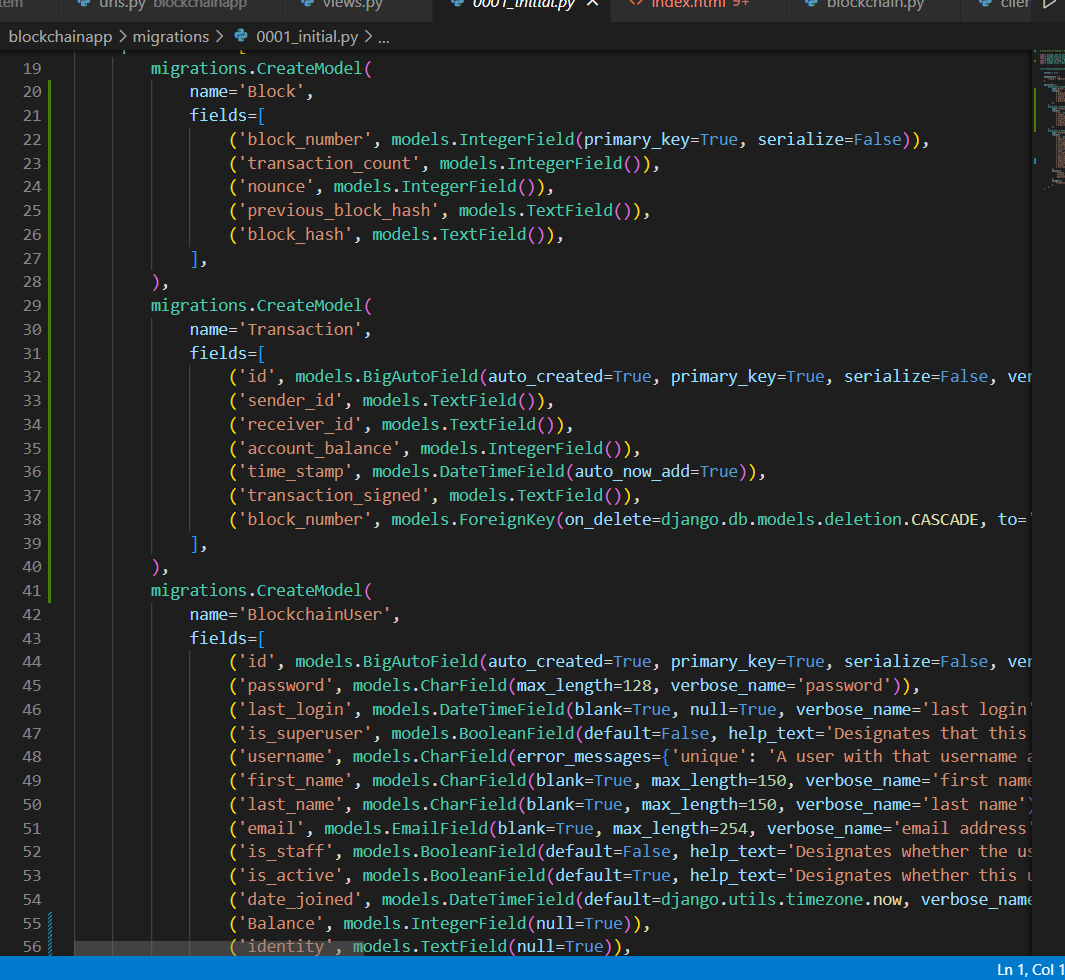
The fig 4.4 diagram shows the code sample for Django request handlers. The request handlers handle rendering of web pages and user requests, this includes the following;

1. **Home Page Handler:** for rendering home page and viewing transaction history
2. **Registration Handler:** for rendering registration page for new user into the budget blockchain system
3. **Account Handler:** for management user account details or information such as account balance, name, private and public key.
4. **Transaction Handler:** This handle the transaction request of users, validation process and confirmation of transactions
5. **Logout Handler:** for handler logout request



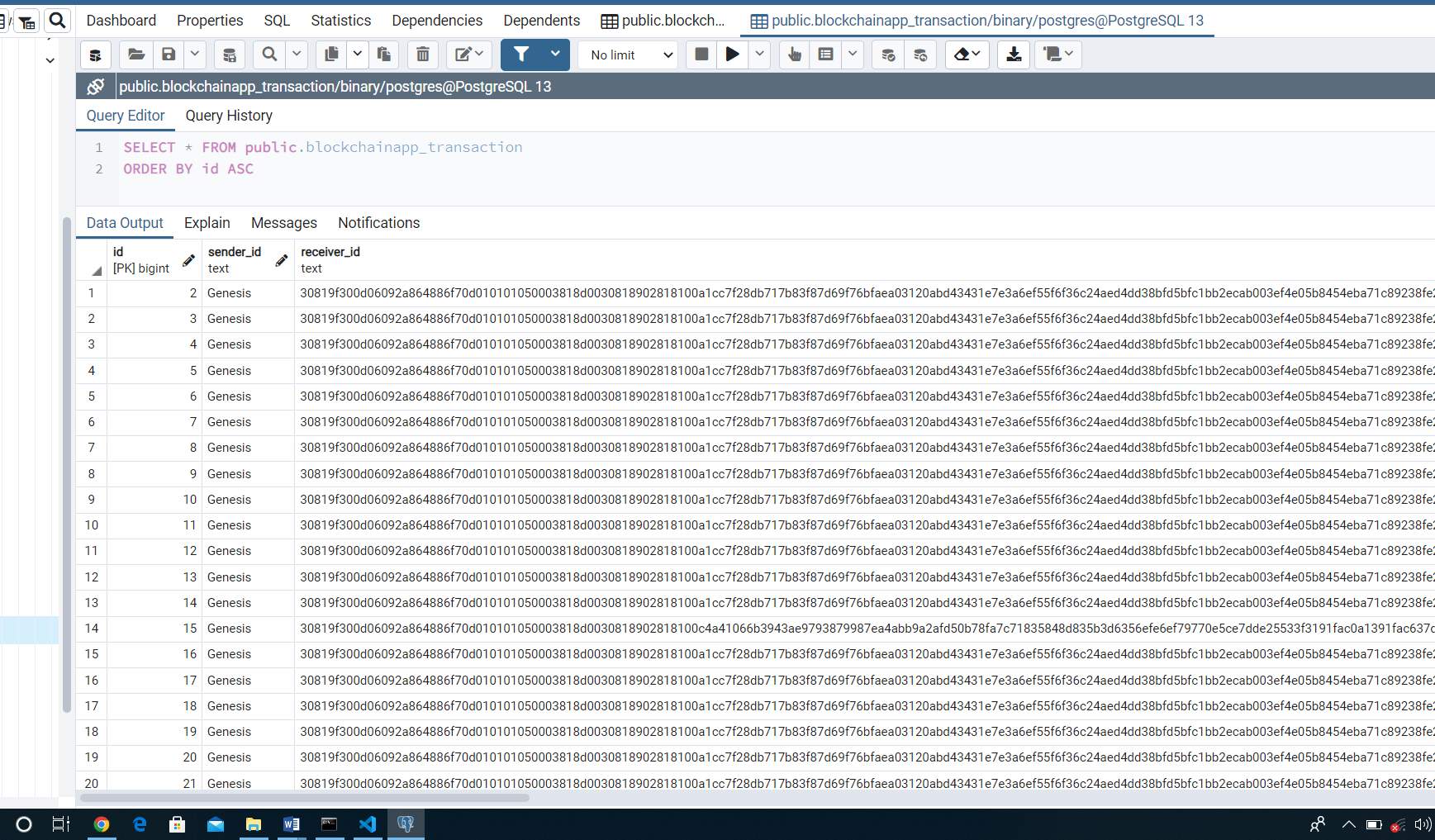
**Fig 4.5 The Django Framework Uniform Resource Locator Mapping**

The Uniform Resource Locator mapping is handling and manage by the Django framework, it connects user request to the appropriate backend code handlers as illustrated in fig 4.5. The URL mapping connect the account request to the account handler, transaction request to the transaction handlers and so on.

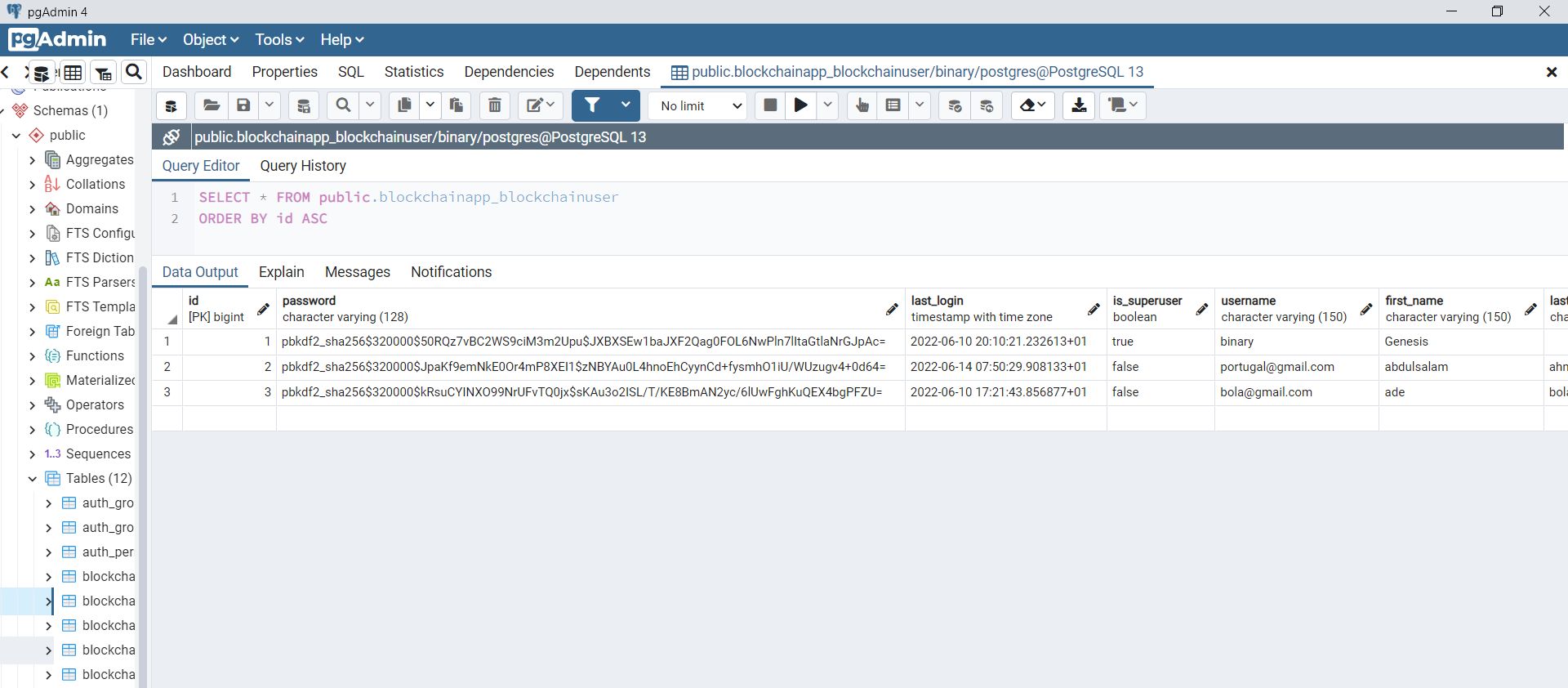


**Fig 4.6 The Migration File for Object (Model) Relational Mapping (ORM)**

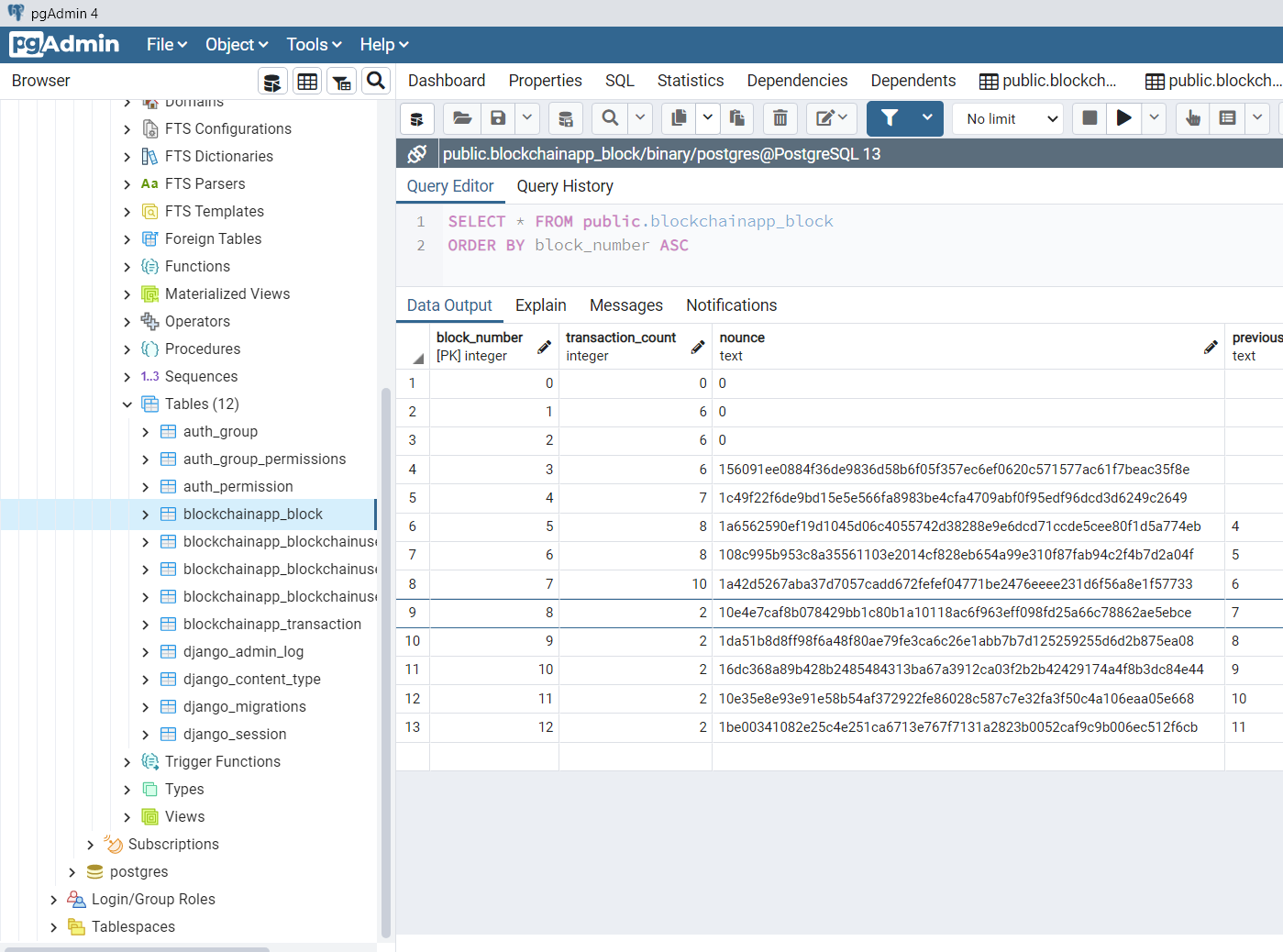
The Fig 4.6 shows the migration file which handles the object (model) to relational (table) mapping. Based on the figure the Block, Transaction and the Blockchain Model are Migrated or converted to a relation table in the Relational Database Management System (RDBMS). The adopted RDBMS in this thesis is PostgreSQL.



**Fig 4.7 PostgreSQL Admin Transaction database record**



**Fig 4.8 PostgreSQL Admin User Account database record**



**Fig 4.9 PostgreSQL Admin Block database record**

Fig 4.7, 4.8, 4.9 illustrate the transaction of user, user account details, and block records in the PostgreSQL pgAdmin4. The pgAdmin4 is an RDBMS database for performing Update, Delete, Add, and Create operation.

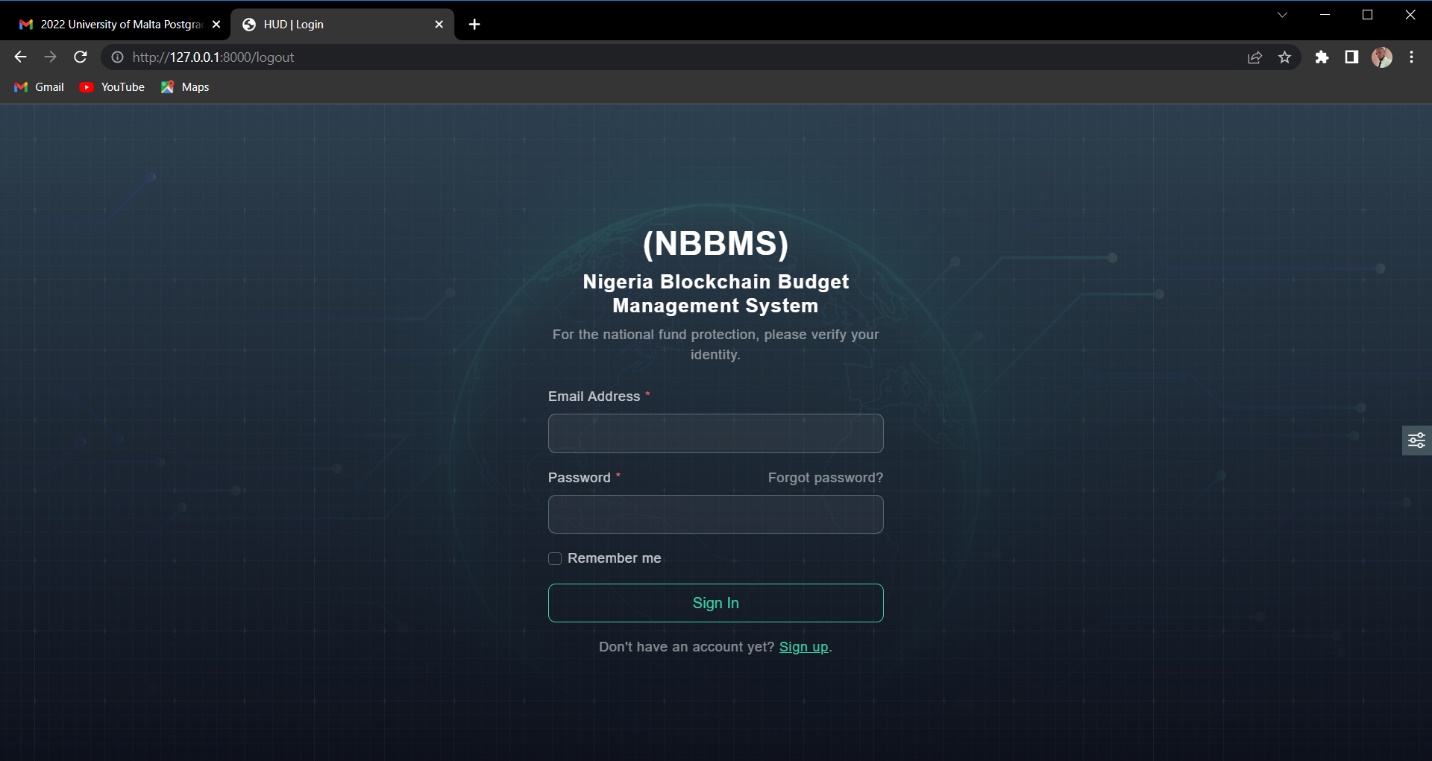
However, the Transaction table contains attribute or columns such as;

1. **Id’s:** to uniquely identify transactions in the table
2. **Sender Id:** Is the hexadecimal representation of the sender public key address.
3. **Receiver Id:** the hexadecimal representation of the receiver public key address.
4. **Value:** holds the amount exchanges between two registered blockchain public address.
5. **Time Stamp:** This holds the actual time the transaction is initiated.
6. **Block Number:** this holds the block number the transaction is attached to, and this attribute a foreign key in the block table.

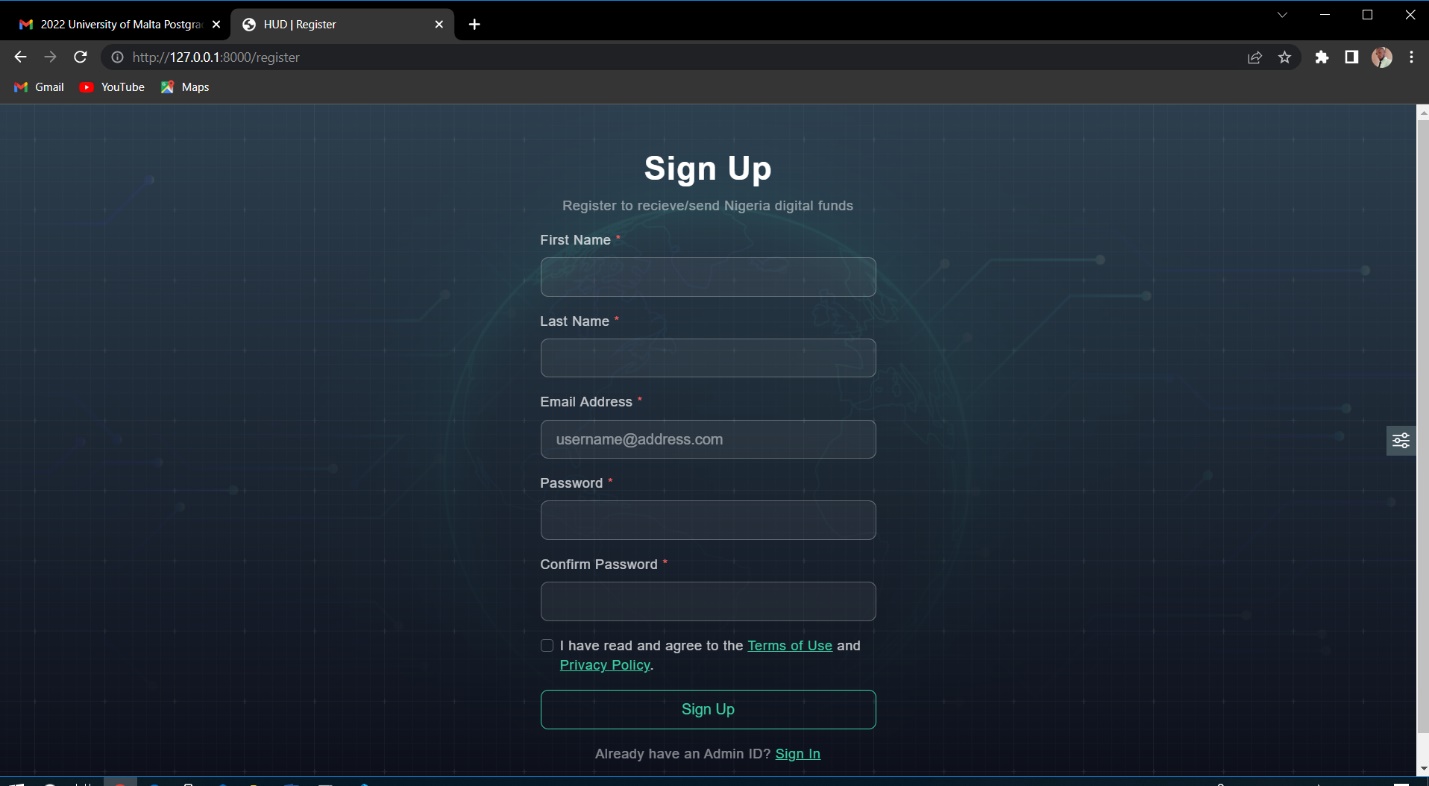
While the Block Table contain attribute such as:

1. **Block Number:** unique id represents each block in the table
2. **Transaction Count:** this contains the number of transaction associated to a particular block.
3. **Nonce:** this hold the nonce value of each block
4. **Previous Block Hash:** the prior block hash value stored in the current block.

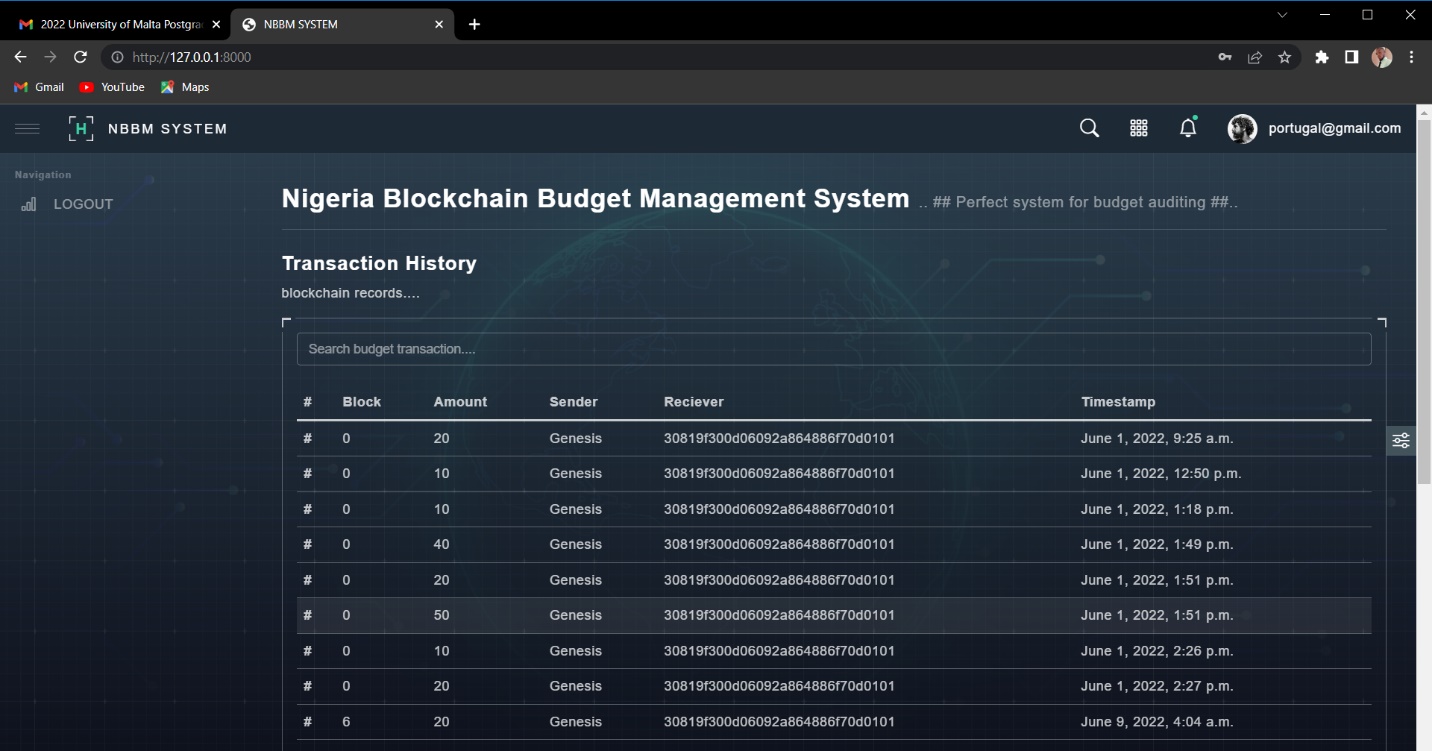
**4.4.3 Django Framework (Template Design)**



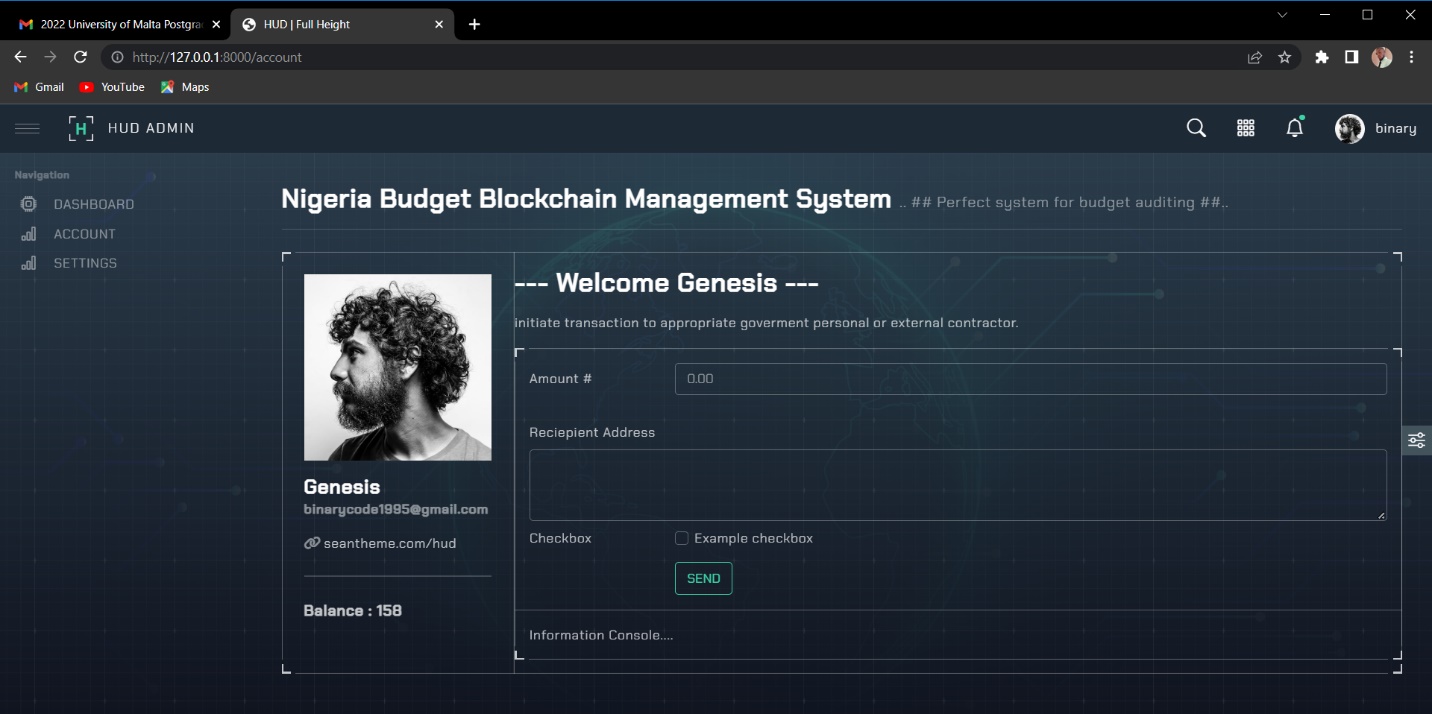
**Fig 4.10 Template design (Login View)**



**Fig 4.11 Template design (Registration View)**



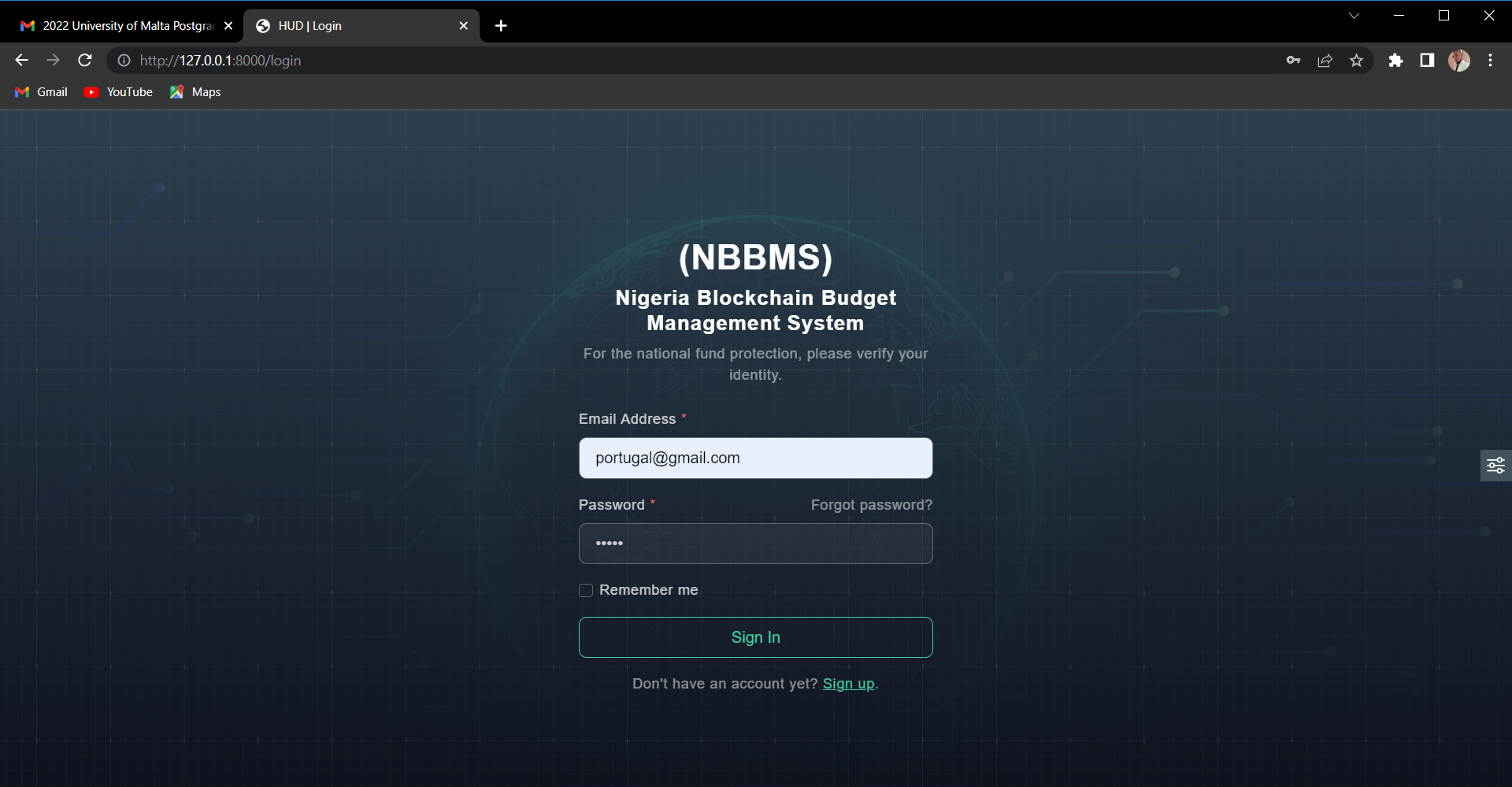
**Fig 4.12 Template design (Home View)**



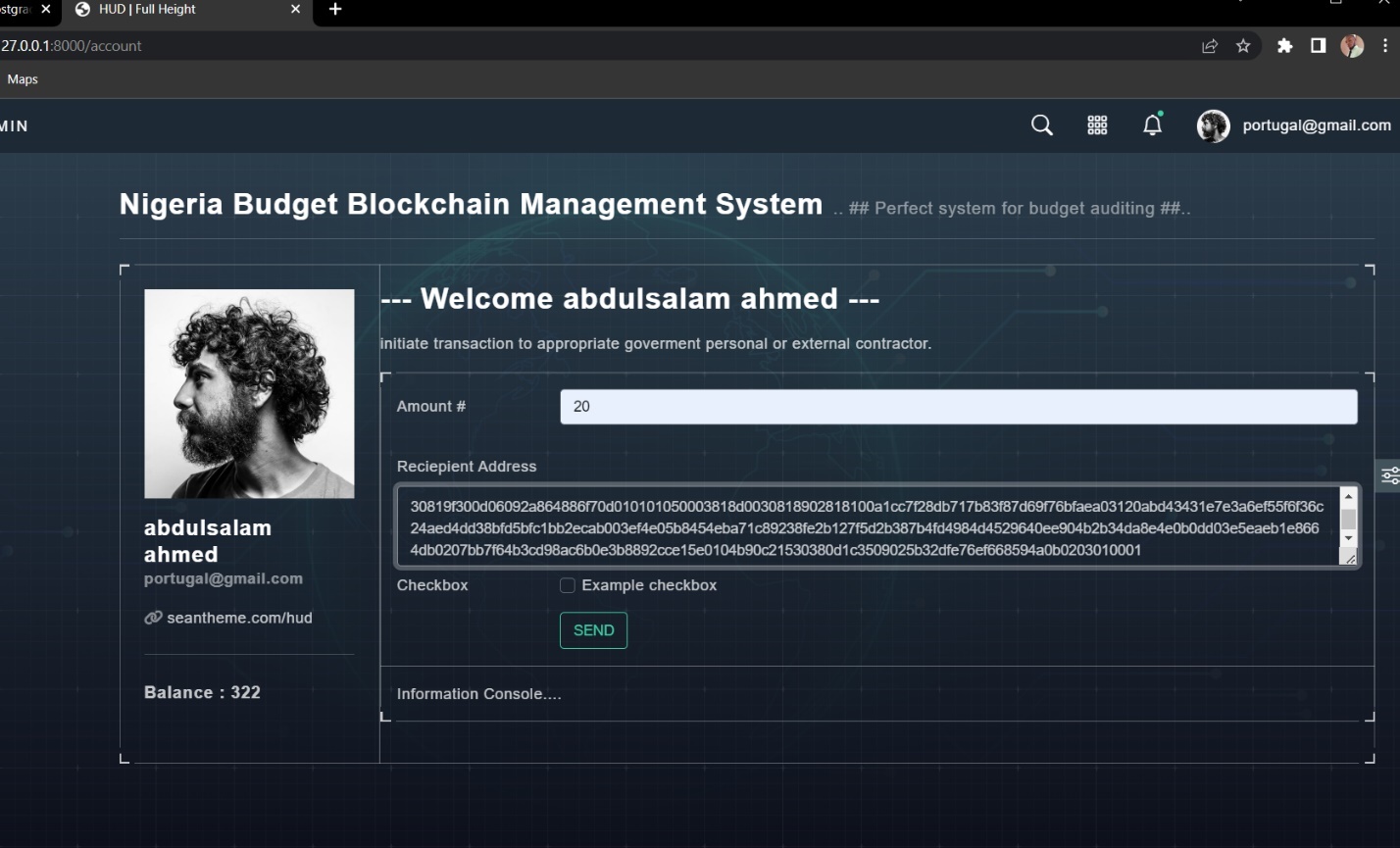
**Fig 4.13 Template design (Account View)**

The fig 4.10, 11, 12 and 13 diagrams show the user interface design. The fig 4.10 shows the login page for accepting user credentials, two text field is provided to users for their email and password details. Moreover, fig 4.11 provide field for registering new user, hence its call registration or sign up page. Information such as username, password, email, first and last name are provided using the provided text fields. Fig 4.12 show the landing pages with transaction history in a tabular form. Finally, fig 4.13 visually show the account section page of each users with their full name, account balance detail and input field for initiating transactions.

**4.5 System Testing, evaluation and discussion**



**Fig 4.14 User login process**



**Fig 4.15 Transaction initiation process**

Based on the fig 4.14 showing a registered users login into the system using a valid credentials, while figure 4.15 illustrate when a valid logged in user enters transactional information such as; valid budget amount (20) to be transferred and a valid recipient address on the blockchain network. On pressing the send button, the request is confirmed if all condition attached to the transaction are meet.

**Evaluation Table 4.2**

|  |  |  |
| --- | --- | --- |
| **Features** | **Traditional Nigeria Budget Management System** **(Ogujiuba & Ehigiamusoe, 2014) (Eigbiremolen et al., 2015), (Aladejare, 2020)** | **Implemented Nigeria blockchain Budget Management System** |
| **Security** | Not Secure due to corrupt government officials | Highly secure due to the blockchain cryptography mechanisms for carrying out transaction |
| **Transparency** | Difficult to trace and maintain. Hence, it’s not completely transparent. | can be easily maintained and its entirely traceable. Hence, its transparent |
| **Decentralized** | Transactions on the traditional budget distribution is controlled by a central authority. Thus, government banks. | transactions are not managed by central authorities. |
| **Integrity** | Budget Transaction stored in the system can be modify or change by any government officials. | Budget transaction Information are completely Immutable (cannot be change), and its permanent (can’t be modify or deleted) |
| **Consensus** | budget funds exchange decision-making is managed by the central authority | The proposed system ensure decision-making by all nodes (users) on the NBBMS |

The evaluation carried out on table 4.2 outline major different between the tradition Nigeria Budget Management System and the proposed system, by using major feature of the blockchain technology which include security, consensus, transparency and decentralized system. However, based on table 4.2 its clearly shown that the propose NBBMS system is more efficient in terms of secure budget distribution of funds, transparent records of transactions, decentralize management of information’s, and easy to manage System.

**Reference**

Aladejare, S. A. (2020). *Are public spending determinants significant in per capita budget spending decisions in Nigeria ?* *June*, 1–15. https://doi.org/10.1002/ijfe.2146

Eigbiremolen, G. time O., Ezema, N. J., & Orji, A. (2015). Dynamics of Budget Deficit and Macroeconomic Fundamentals: Further Evidence from Nigeria. *International Journal of Academic Research in Business and Social Sciences*, *5*(5). https://doi.org/10.6007/ijarbss/v5-i5/1590

Ogujiuba, K. K., & Ehigiamusoe, K. U. (2014). *Capital Budget Implementation in Nigeria : Evidence from the 2012 Capital Capital Budget Implementation in Nigeria : Evidence from the 2012 Capital Budget*. *September*. https://doi.org/10.5709/ce.1897-9254.147

**Chapter Five**

**Summary, Conclusion and Recommendation**

* 1. **Introduction**

this section entails the entire project summarization, the conclusion statement and the recommendation for future works

* 1. **Summary**

In terms of preparation and implementation, Nigeria's budgeting process is still problematic. In today's governmental operations, severe obstacles are highlighted, particularly among government officials who lack the institutional background, experience, intelligence, and understanding required to achieve the necessary skills for holding the government accountable. The government's agencies are dealing with challenges of tracking audits, accountability, and responsibility among government officials, as well as monetary circulation inside the country. As a result, it's critical to create a framework or system that can effectively control the budget's effective and efficient use during the planning and execution stages. This research paper, on the other hand, offered a decentralized system (blockchain technology) for budget management, budget auditing, accountability, responsibility, building trust, and untampered budget transactions. The evaluation focused on the key differences between the traditional Nigerian budget management system and the suggested system, leveraging key features of blockchain technology such as security, consensus, transparency, and decentralization.

* 1. **Conclusion**

However, based on the evaluation, the proposed NBBMS system is definitely more efficient in terms of secure budget allocation of money, transparent transaction records, decentralized information management, and ease of management.

* 1. **Recommendation**

The proposed system can be recommended for any developing country, most specifically for corrupt countries. Budget allocations can be efficiently distributed to the appropriate section, and information of transaction are secure and unmodifiable for future auditing.